

MOEA/D-GM: Using probabilistic graphical models in MOEA/D for solving combinatorial optimization problems

Murilo Zangari de Souza¹, Roberto Santana², Aurora Trinidad Ramirez Pozo¹, Alexander Mendiburu²

¹DInf - Federal University of Paraná, CP: 19081, CEP 19031-970, Curitiba, Brazil

²Intelligent Systems Group, University of the Basque Country UPV/EHU, San Sebastián, Spain

Email: {murilo.zangari, auroratrinidad}@gmail.com, {roberto.santana, alexander.mendiburu}@ehu.eus

Abstract—Evolutionary algorithms based on modeling the statistical dependencies (interactions) between the variables have been proposed to solve a wide range of complex problems. These algorithms learn and sample probabilistic graphical models able to encode and exploit the regularities of the problem. This paper investigates the effect of using probabilistic modeling techniques as a way to enhance the behavior of MOEA/D framework. MOEA/D is a decomposition based evolutionary algorithm that decomposes an multi-objective optimization problem (MOP) in a number of scalar single-objective subproblems and optimizes them in a collaborative manner. MOEA/D framework has been widely used to solve several MOPs. The proposed algorithm, MOEA/D using probabilistic Graphical Models (MOEA/D-GM) is able to instantiate both univariate and multi-variate probabilistic models for each subproblem. To validate the introduced framework algorithm, an experimental study is conducted on a multi-objective version of the deceptive function *Trap5*. The results show that the variant of the framework (MOEA/D-Tree), where tree models are learned from the matrices of the mutual information between the variables, is able to capture the structure of the problem. MOEA/D-Tree is able to achieve significantly better results than both MOEA/D using genetic operators and MOEA/D using univariate probability models in terms of approximation to the *true* Pareto front.

Index Terms—multi-objective optimization, MOEA/D, probabilistic graphical models, deceptive functions, EDAs

I. INTRODUCTION

Several real-world problems can be stated as multi-objective optimization problems (MOPs) which have two or more objectives to be optimized. Very often these objectives conflict with each other. Therefore, no single solution can optimize these objectives at the same time. Pareto optimal solutions are of very practical interest to decision makers for selecting a final preferred solution. Most MOPs may have many or even infinite optimal solutions and it is very time consuming (if not impossible) to obtain the complete set of optimal solutions [1].

Since early nineties, much effort has been devoted to develop evolutionary algorithms for solving MOPs. Multi-objective evolutionary algorithms (MOEAs) aim at finding a set of representative Pareto optimal solutions in a single run [1]–[3].

Different strategies have been used as criteria to maintain a population of optimal non-dominated solutions, or Pareto set (*PS*), and consequently finding an approximated Pareto front

(*PF*). Popular strategies include:¹ (i) Pareto dominance based; (ii) Indicator based and (iii) Decomposition based (also called, scalarization function based) [3], [5].

Zhang and Li [2] proposed a decomposition based algorithm called MOEA/D (multi-objective Evolutionary Algorithm Based on Decomposition) framework, which decomposes a MOP into a number of single-objective scalar optimization subproblems and optimizes them simultaneously in a collaborative manner using the concept of neighborhood between the subproblems.

Current research on MOEA/D are various and include the extension of these algorithms to continuous MOPs with complicated Pareto sets [6], many-objective optimization problems [5], [7], [8], methods to parallelize the algorithm [9], incorporation of preferences to the search [10], automatic adaptation of weight vectors [11], new strategies of selection and replacement to balance convergence and diversity [3], hybridization with local searches procedures [12], [13], etc.

However, it has been shown that traditional EA operators fail to properly solve some problems when certain characteristics are present in the problem such as deception [14]. A main reason for this shortcoming is that, these algorithms do not consider the dependencies between the variables of the problem [14]. To address this issue, evolutionary algorithms that (instead of using classical genetic operators) incorporate machine learning methods have been proposed. These algorithms are usually referred to as Estimation of Distribution Algorithms (EDAs). In EDAs, the collected information is represented using a probabilistic model which is later employed to generate (sample) new solutions [14]–[16].

EDAs based on modeling the statistical dependencies between the variables of the problem have been proposed as a way to encode and exploit the regularities of complex problems. Those EDAs use more expressive probabilistic models that, in general, are called probabilistic graphical models (PGMs) [14]. A PGM comprises a graphical component representing the conditional dependencies between the variables, and a set of parameters, usually tables of marginal and conditional probabilities. Additionally, the analysis of the

¹There exist algorithms, such as NSGAIII [4], that combine the ideas from both decomposition based and Pareto dominance based.

graphical components of the PGMs learned during the search can provide information about the problem structure. MOEDAs using PGMs have been also applied for solving different MOPs [17]–[19].

In the specified literature of MO-EDAs, a number of algorithms that integrate to different extents the idea of probabilistic modeling into MOEA/D have been previously proposed [20]–[24]. Usually, these algorithms learn and sample a probabilistic model for each subproblem. Most of them use only univariate models, which are not able to represent dependencies between the variables.

In this paper, we investigate the effect of using probabilistic modeling techniques as a way to enhance the behavior of MOEA/D. We propose a MOEA/D framework able to instantiate different PGMs. The general framework is called MOEA/D Graphical Models (MOEA/D-GM). The goals of this paper are: (i) Introduce MOEA/D-GM as a class of MOEA/D algorithms that learn and sample probabilistic graphical models defined on discrete domain; (ii) To empirically show that MOEA/D-GM can improve the results of traditional MOEA/D for deceptive functions; (iii) To investigate the influence of modeling variables dependencies in different metrics used to estimate the quality of the Pareto front approximations and (iv) to show, for the first time, evidence of how the problem structure is captured in the models learned by MOEA/D-GM.

The paper is organized as follows. In the next section some relevant concepts used in the paper are introduced. Sections III and IV respectively explain the basis of MOEA/D and the EDAs used in the paper. Section V introduces the MOEA/D-GM and explains the enhancements required by EDAs in order to efficiently work within the MOEA/D context. In Section VI we discuss the class of functions which are the focus of this paper, multi-objective deceptive functions. We explain the rationale of applying probabilistic modeling for these functions. Related work is discussed in detail in Section VII. The experiments that empirically show the behavior of MOEA/D-GM are described in Section VIII. The conclusions of our paper and some possible trends for future work are presented in Section IX.

II. PRELIMINARIES

Let $\mathbf{X} = (X_1, \dots, X_n)$ denote a vector of discrete random variables. $\mathbf{x} = (x_1, \dots, x_n)$ is used to denote an assignment to the variables and $\mathbf{x} \in \{0, 1\}^n$. A population is represented as a set of vectors x^1, \dots, x^N where N is the size of the population. Similarly, x_j^i represents the assignment to the j th variable of the i th solution in the population.

A general MOP can be defined as follows [1]:

$$\begin{aligned} \text{minimize (or maximize)} \quad & F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ \text{subject to} \quad & \mathbf{x} \in \Omega \end{aligned} \quad (1)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ is the decision variable vector of size n , Ω is the *decision space*, $F : \Omega \rightarrow R^m$ consists of m objective functions and R^m is the *objective space*.

Pareto optimality is used to define the set of optimal solutions.

Pareto optimality² [1]: Let $\mathbf{x}, \mathbf{y} \in \Omega$, \mathbf{x} is said to *dominate* \mathbf{y} if and only if $f_l(\mathbf{x}) \leq f_l(\mathbf{y})$ for all $l \in \{1, \dots, m\}$ and $f_l(\mathbf{x}) < f_l(\mathbf{y})$ for at least one l . A solution $\mathbf{x}^* \in \Omega$ is called *Pareto optimal* if there is no other $\mathbf{x} \in \Omega$ which *dominates* \mathbf{x}^* . The set of all the Pareto optimal solutions is called the *Pareto set (PS)* and the solutions mapped in the *objective space* are called *Pareto front (PF)*, i.e., $PF = \{F(\mathbf{x}) | \mathbf{x} \in PS\}$. In many real life applications, the *PF* is of great interest to decision makers for understanding the trade-off nature of the different objectives and selecting their preferred final solution.

III. MOEA/D

MOEA/D [2] decomposes a MOP into a number (N) of scalar single objective optimization subproblems and optimizes them simultaneously in a collaborative manner using the concept of neighborhood between the subproblems. Each subproblem i is associated to a weight vector $\lambda^i = (\lambda_1^i, \dots, \lambda_m^i)^T$ and the set of all weight vectors is $\{\lambda^1, \dots, \lambda^N\}$.

Several decomposition approaches have been used in the MOEA/D [25]. The *Weighted Sum* and *Tchebycheff* are two of the most common approaches used, mainly in combinatorial problems [2], [25].

Let $\lambda = (\lambda_1, \dots, \lambda_m)^T$ be a weight vector, where $\sum_{l=1}^m \lambda_l = 1$ and $\lambda_l \geq 0$ for all $l = 1, \dots, m$.

Weighted Sum Approach: The optimal solution to the following scalar single-optimization problem is defined as:

$$\begin{aligned} \text{minimize (or maximize)} \quad & g^{ws}(\mathbf{x}|\lambda) = \sum_{l=1}^m \lambda_l f_l(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in \Omega \end{aligned} \quad (2)$$

where we use $g^{ws}(\mathbf{x}|\lambda)$ to emphasize that λ is a weight vector in this objective function. \mathbf{x} is a Pareto optimal solution to (1) if the *PF* of (1) is convex (or concave).

Tchebycheff approach: The optimal solution to the following scalar single-optimization problems is defined as:

$$\begin{aligned} \text{minimize} \quad & g^{te}(\mathbf{x}|\lambda, z^*) = \max_{1 \leq l \leq m} \{\lambda_l |f_l(\mathbf{x}) - z_l^*|\}; \\ \text{subject to} \quad & \mathbf{x} \in \Omega \end{aligned} \quad (3)$$

where $z^* = (z_1^*, \dots, z_m^*)^T$ is the reference point, i.e., $z_l^* = \max\{f_l(\mathbf{x}) | \mathbf{x} \in \Omega\}$ for each $l = 1, \dots, m$. For each Pareto optimal solution x^* there exists a weight vector λ such that x^* is the optimal solution of (3) and each optimal solution of (3) is Pareto optimal of (1).

The neighborhood relation among the subproblems is defined in MOEA/D. The neighborhood for each subproblem i , $B(i) = \{i_1, \dots, i_T\}$, is defined according to the Euclidean distance between its weight vector and the other weight vectors. The relationship of the neighbor subproblems is used for the selection of parent solutions and the replacement of old solutions (also called update mechanism). The size of the neighborhood used for selection and replacement plays a vital role in MOEA/D to exchange information among the subproblems [3], [5]. Moreover, optionally, an external population *EP* is used to maintain all Pareto optimal solutions found during

²This definition of domination is for minimization. For maximization, the inequalities should be reversed.

the search. Algorithm 1 (adapted from [3]) presents the pseudo-code of the general MOEA/D which serves as a basis for this paper.

Algorithm 1: General MOEA/D framework

```

1  [Pop, Ts, Tr, EP] ← Initialization() // Pop is the initial
   population of size N. Set B(i) = Ts and R(i) = Tr, where
   B(i) and R(i) are, respectively, the neighborhood size for
   selection and replacement. EP is the external population.
2  While a termination condition not met
3    For each subproblem i ∈ 1, ..., N at each generation
4      y ← Variation(B(i))
5      Evaluate F(y) using the fitness function
6      Update_Population(Pop, y, R(i))
7      Update_EP(F(y), EP)
8  Return Pop, EP

```

1) *Initialization*: The N weight vectors $\lambda^1, \dots, \lambda^N$ are set. The Euclidean distance between any two weight vectors is computed. For each subproblem i , the set of neighbors for the selection step $B(i)$ and the update step $R(i)$ are initialized with the T_s and T_r closest neighbors respectively according to the Euclidean distance. The initial population $Pop = x^1, \dots, x^N$ is generated in a random way and their corresponding fitness function $F(x^1), \dots, F(x^N)$ are computed. The external Pareto EP is initialized with the non-dominated solutions from the initial population Pop .

2) *Variation*: The reproduction is performed using $B(i) = \{i_1, \dots, i_{T_s}\}$ to generate a new solution y . The conventional algorithm selects two parent solutions from $B(i)$ and applies *crossover* and *mutation* to generate y .

3) *Update Population*: This step decides which subproblems should be updated from $R(i) = \{i_1, \dots, i_{T_r}\}$. The current solutions of these subproblems are replaced by y if y has a better aggregation function value $g^{te}(y|\lambda^r)$ than $g^{te}(x^r|\lambda^r)$, $x^r \in R(i)$.

4) *Update EP*: This step removes from EP the solutions dominated by y and adds y if no solution dominates it.

A. Improvements on MOEA/D framework

Wang et. al [3] have reported that different problems need different trade-offs between diversity and convergence, which can be controlled by the different mechanisms and parameters of the algorithm. As so far, most proposed MOEA/D versions adopted the $(\mu + 1)$ -selection/variation scheme, which selects μ individuals from a population and generates 1 offspring [3]. Different strategies for selection and replacement have been proposed [3], [5]. In this paper, the replacement proposed by [6] is used. In this strategy, the maximal number of solutions replaced by a new solution y is bounded by n_r , which should be set to be much smaller than T_r . This replacement mechanism prevents one solution having many copies in the population.

IV. ESTIMATION OF DISTRIBUTION ALGORITHMS

EDAs [14]–[16] are stochastic population-based optimization algorithms that explore the space of candidate solutions

by sampling a probabilistic model constructed from a set of selected solutions found so far. Usually, in EDAs, the population is ranked according to the fitness function. From this ranked population, a subset of the most promising solutions are selected by a *selection operator* (such as: *Truncation selection* with truncation threshold $t = 50\%$). The algorithm then constructs a probabilistic model which attempts to estimate the probability distribution of the selected solutions. Then, according to the probabilistic model, new solutions are sampled and incorporated into the population, which can be entirely replaced [26]. The algorithm stops when a termination condition is met such as the number of generations.

We work with positive distributions denoted by p . $p(x_I)$ denotes the marginal probability for $\mathbf{X}_I = \mathbf{x}_I$. $p(x_j | x_k)$ denotes the conditional probability distribution of $X_j = x_j$ given $X_k = x_k$. The set of selected promising solutions is represented as S . Algorithm 2 presents the general EDA procedure.

Algorithm 2: General EDA

```

1  Pop ← Generate N solutions randomly
2  While a termination condition not met
3    For each solution xi compute its fitness function F(xi)
4    Select the most promising solutions S from Pop
5    Build a probabilistic model M of solutions in S
6    Generate new candidate solutions sampling from M and
   add them to Pop

```

The way in which the learning and sampling components of the algorithm are implemented are also critical for their performance and computational cost [14]. In the next section, the univariate and *tree-based* probabilistic models are introduced.

A. Univariate probabilistic models

In the univariate marginal distribution (or univariate probabilistic model), the variables are considered to be independent, and the probability of a solution is the product of the univariate probabilities for all the variables:

$$p_u(\mathbf{x}) = \prod_{j=1}^n p(x_j) \quad (4)$$

One of the simplest EDAs that use the univariate model is the univariate marginal distribution algorithm (UMDA) [27]. UMDA uses a probability vector $p_u(\mathbf{x})$ as the probabilistic model, where $p(x_j)$ denotes the univariate probability associated to the corresponding discrete value. In this paper we focus on binary problems. To learn the probability vector for these problems, each $p(x_j)$ is set to the proportion of "1s" in the selected population S . To generate new solutions, each variable is independently sampled. Random values are assigned to the variables and following the corresponding univariate distribution.

The population-based incremental learning (PBIL) [28], like UMDA, uses the probabilistic model in the form of a probability vector $p_u(\mathbf{x})$. The initial probability of a "1" in each position $p(x_j)$ is set to 0.5. The probability vector is

updated using each selected solution \mathbf{x} in S . For each variable, the corresponding entry in the probability vector is updated by:

$$p(x_j) = (1.0 - \alpha)p(x_j) + (\alpha * x_j) \quad (5)$$

where α is the learning rate specified by the user.

To prevent premature convergence, each position of the probability vector is slightly varied at each generation based on a mutation rate parameter [26], [28]. Recently, it has been acknowledged [29] that implementations of the probabilistic vector update mechanism in PBIL are in fact different, and produce an important variability in the behavior of PBIL.

Univariate approximations are expected to work well for functions that can be additively decomposed into functions of order one (e.g. $g(\mathbf{x}) = \sum_j x_j$). Also, other non additively decomposable functions can be solved with EDAs that use univariate models (e.g. $g(\mathbf{x}) = \prod_j x_j + \sum_j x_j$) [30].

B. Tree-based models

Different from the EDAs that use univariate models, some EDAs can assume dependencies between the decision variables. In this case, the probability distribution is represented by a PGM.

Tree-based models [31] are PGMs capable to capture some *pairwise* interactions between variables. In a tree model, the conditional probability of a variable may only depend on at most one other variable, its parent in the tree.

The probability distribution $p_{\mathcal{T}}(\mathbf{x})$ that is conformal with a tree model is defined as:

$$p_{\mathcal{T}}(\mathbf{x}) = \prod_{j=1}^n p(x_j | pa(x_j)), \quad (6)$$

where $pa(X_j)$ is the parent of X_j in the tree, and $p(x_j | pa(x_j)) = p(x_j)$ when $pa(X_j) = \emptyset$, i.e. X_j is a root of the tree.

The bivariate marginal distribution algorithm (BMDA) proposed in [31] uses a model based on a set of mutually independent trees (a forest)³. In each iteration of the algorithm, a tree model is created and sampled to generate new candidate solutions based on the conditional probabilities learned from the population.

The algorithm Tree-EDA proposed in [32] combines features from algorithms introduced in [33] and [31]. In Tree-EDA, the learning procedure works as follows:

- Step 1: Compute the univariate and bivariate marginal frequencies $p_j(x_j)$ and $p_{jk}(x_j|x_k)$ using the set of selected promising solutions S ;
- Step 2: Calculate the matrix of mutual information using the univariate and bivariate frequencies;
- Step 3: Calculate the maximum weight spanning tree from the mutual information. Compute the parameters of the model.

The idea is that by computing the maximum weight spanning tree from the matrix of mutual information the algorithm

will be able to capture the most relevant bivariate dependencies between the problem variables.

Details on EDAs that use the tree models can be obtained from [31], [33]. More details on the use of PGMs for probabilistic modeling in EDAs can be obtained from [14].

C. Optimal Mutation Rate for EDAs

Most of EDAs do not use any kind of stochastic mutation. However, for certain problems, lack of diversity in the population is a critical issue that can cause EDAs to produce poor results due to premature convergence. As a remedy to this problem, in [34], the authors propose the use of Bayesian *priors* as an effective way to introduce a mutation operator into UMDA. Bayesian priors are used for the computation of the univariate probabilities in such a way that the computed probabilities will include a mutation-like effect. In this paper we use the Bayesian prior as a natural way to introduce mutation in the MOEA/D-GM. In the following this strategy is described.

There are two possibilities to estimate the probability of "head" of a biased coin. The maximum-likelihood estimate counts the number of occurrences of each case. With m times "head" in l throws, the probability is estimated as $p = \frac{m}{l}$. The Bayesian approach assumes that the probability of "head" of a biased coin is determined by an unknown parameter θ . Starting from an *a priori* distribution of this parameter. Using the Bayesian rule the univariate probability is computed as $p = \frac{m+r}{l+2r}$, and the so called *hyperparameter* r has to be chosen in advance.

To relate the Bayesian *prior* to the mutation, the authors used the following theorem: *For binary variables, the expectation value for the probability using a Bayesian prior with parameter r is the same as mutation with mutation rate $\mu = \frac{r}{l+2r}$ and using the maximum likelihood estimate [34].*

V. MOEA/D-GM: AN MOEA/D ALGORITHM WITH GRAPHICAL MODELS

Algorithm 3 shows the steps of the proposed MOEA/D-GM framework. First, the initialization procedure generates N initial solutions and the external Pareto (*EP*) is initialized with the non-dominated solutions. Within the main while-loop, in case the termination criteria are not met, for each subproblem i , a probabilistic model is learned using as base population all solutions in the neighborhood $B(i)$. Then, the sampling procedure is used to generate a new solution from the model. The new solutions is used to update the parent population according to an elite-preserving mechanism. Finally, the new solution is used to update *EP* as described in Algorithm 1.

A simple way to use probabilistic models into MOEA/D is learning and sampling a probabilistic model for each scalar subproblem i using the set of closest neighbors as the selected population. Therefore, at each generation, the MOEA/D-GM keeps N probabilistic models.

The EDAs presented in the previous section: (i) UMDA [27], (ii) PBIL [28] and (iii) Tree-EDA [32] can be instantiated in the framework. Also, the genetic operators (*crossover* and

³For convenience of notation, this set of mutually independent trees is referred as Tree.

Algorithm 3: MOEA/D-GM

```

1   $[Pop, T_s, T_r, EP] \leftarrow \text{Initialization}()$  //  $Pop$  is the initial population
   of size  $N$  that is randomly generated. Set  $B(i) = T_s$  and  $R(i) = T_r$ ,
   where  $B(i)$  and  $R(i)$  are, respectively, the neighborhood size for selection
   and replacement.  $EP$  is the external population.
2  While a termination condition not met
3    For each subproblem  $i \in 1, \dots, N$  at each generation
4      Learning()
5        Case GA: Choose two parent solutions  $x^1 x^2$  from  $B(i)$ 
6        Case UMDA: Learn a probabilistic vector  $p^i$  using the solutions
           from  $B(i)$ 
7        Case PBIL: Learn an incremental probabilistic vector  $p^i$  using the
           solutions from  $B(i)$ 
8        Case Tree-EDA: Learn a tree model  $p^i_{\mathcal{T}}$  using the solutions from
            $B(i)$ 
9      Sampling() // Try up to  $T_s$  times to sample a new solution  $\mathbf{y}$  that is
           different from any solution from  $B(i)$ 
10     Case GA: Apply the crossover and mutation to generate  $\mathbf{y}$ 
11     Case PBIL or UMDA: Sample  $\mathbf{y}$  using the probability vector  $p^i$ 
12     Case Tree-EDA: Sample  $\mathbf{y}$  using the tree model  $p^i_{\mathcal{T}}$ 
13     Compute the fitness function  $F(\mathbf{y})$ 
14     Update  $Population(Pop, \mathbf{y}, R(i))$ 
15     Update  $EP(F(\mathbf{y}), EP)$ 
16  Return  $Pop, EP$ 

```

mutation), that are used in the standard MOEA/D, can be applied using the learning and sampling procedures.

The general MOEA/D-GM allows the introduction of other types of probabilistic graphical models (PGMs) like Bayesian networks [35] and Markov networks [36]. Different classes of PGMs can be also used for each scalar subproblem but we do not consider this particular scenario in the analysis presented here.

Moreover, the MOEA/D-GM introduces some particular features that are explained in the next section.

1) Learning models from the complete neighborhood:

Usually in EDAs, population sizes are large and the size of the selected population can be as high as the population size. In MOEA/D, each subproblem has a set of neighbors that plays a similar role to the selected population⁴. Therefore, the main difference between the MOEA/D-GM and the EDAs presented in section IV is that, in the former, instead of selecting a subset of individuals based on their fitness to keep a unique probabilistic model, a probabilistic model is computed over the neighborhood solutions of each scalar subproblem.

2) *Diversity preserving sampling*: In preliminary experiments we detected that one cause for early convergence of the algorithm was that solutions that were already in the population were newly sampled. Sampling solutions already present in the population is also detrimental in terms of efficiency since these solutions have to be evaluated. As a way to avoid this situation, we added a simple procedure in which each new sampled solution is tested for presence in the neighborhood $B(i)$ of each subproblem i .

If a new sampled solution \mathbf{y} is equal to a parent solution from $B(i)$ then the algorithm discards the solution and samples a new one until a different solution has been found or a

⁴Usually, the size of the neighborhood is $N \times 0.1$. Although, as in [6], the propose MOEA/D has a low probability to select two parent solutions from the complete population.

maximum number of trials is reached. This procedure is specially suitable to deal with expensive fitness functions. The maximum number of tries can be specified by the user. We call the sampling that incorporates this verification procedure as *diversity preserving sampling* (*ds*). When it is applied on MOEA/D-GM, to emphasize it, the algorithm is called MOEA/D-GM-ds.

VI. MULTI-OBJECTIVE DECEPTIVE OPTIMIZATION PROBLEM

There exists a class of scalable problems where the difficulty is given by the interactions that arise among subsets of decision variables. Thus, some problems should require the algorithm to be capable of linkage learning, i.e., identifying and exploring interactions between the decision variables to provide effective exploration. Decomposable deceptive problems [37] have played a fundamental role in the analysis of EAs. As mentioned before, one of the advantages of EDAs that use probabilistic graphical models is their capacity to capture the structure of deceptive functions. Different works in the literature have proposed EDAs [38]–[41] and MO-EDAs [42], [43] for solving decomposable deceptive problems.

One example of this class of decomposable deceptive functions is the *Trap-k*, where k is the fixed number of variables in the subsets (also called, partitions or *building blocks*) [38]. Traps deceive the algorithm away from the optimum if interactions between the variables in each partition are not considered. According to [42], that is why standard crossover operators of genetic algorithms fail to solve traps unless the bits in each partition are located close to each other in the chosen representation.

Pelikan et al [42] used a bi-objective version of *Trap-k* for analyzing the behavior of multi-objective hierarchical BOA (hBOA).

The functions f_{trap5} (Equation (7)) and f_{inv_trap5} (Equation (8)) consist in evaluating a vector of decision variables $\mathbf{x} \in \{0, 1\}^n$, in which the positions of \mathbf{x} are divided into disjoint subsets or partitions of 5 bits each (n is assumed to be a multiple of 5). The partition is fixed during the entire optimization run, but the algorithm is not given information about the partitioning in advance. Bits in each partition contribute to *trap* of order 5 using the following functions:

$$f_{trap5}(\mathbf{x}) = \sum_{k=0}^{l-1} \sum_{i=1}^5 trap5(x_{5*k+i}) \quad (7)$$

$$trap5(u) = \begin{cases} 5 & \text{if } u = 5 \\ 4 - u & \text{if } u < 5 \end{cases}$$

$$f_{inv_trap5}(\mathbf{x}) = \sum_{k=0}^{l-1} \sum_{i=1}^5 inv_trap5(x_{5*k+i}) \quad (8)$$

$$inv_trap5(u) = \begin{cases} 5 & \text{if } u = 0 \\ u - 1 & \text{if } u > 0 \end{cases}$$

where l is the number of building blocks, i.e., $n = 5l$, and u is the number of ones in the input string of 5 bits.

In the bi-objective problem, f_{trap5} and f_{inv_trap5} are conflicting. Thus, there is not one single global optimum solution but a set of Pareto optimal solutions. Moreover, the amount of possible solutions grows exponentially with the problem size [39]. Multi-objective *Trap-k* has been previously investigated for Pareto based multi-objective EDAs [42]–[44].

VII. RELATED WORK

In this section we review a number of related works emphasizing the differences to the work presented in this paper. Besides, we did not find any previous report of multi-objective decomposition approaches that incorporate PGMs for solving combinatorial MOPs.

A. MOEA/D using univariate EDAs

In [21], the multi-objective estimation of distribution algorithm based on decomposition (MEDA/D) is proposed for solving the multi-objective Traveling Salesman Problem (mTSP). For each subproblem i , MEDA/D uses a matrix Q^i to represent the connection "strength" between cities in the best solutions found so far. Matrix Q^i is combined with a priori information about the distances between cities of problem i , in a new matrix P^i that represents the probability that the s th and t th cities are connected in the route of the i th sub-problem. Although each matrix P^i encodes a set of probabilities relevant for each corresponding TSP subproblem, these matrices can not be considered PGMs since they do not comprise a graphical component representing the dependencies of the problem. The type of updates applied to the matrices is more related to parametric learning than to structural learning as done in PGMs. Furthermore, this type of "models" resemble more the class of structures traditionally used for ACO and they heavily depend on the use of prior information (in the case of MEDA/D, the incorporation of information about the distances between cities). Therefore, we do not consider MEDA/D as a member of the MOEA/D-GM class of algorithms.

Another approach combining the use of probabilistic models and MOEA/D for mTSP is presented in [20]. In that paper, a univariate model is used to encode the probabilities of each city of the TSP being assigned to each of the possible positions of the permutation. Therefore, the model is represented as a matrix of dimension $n \times n$ comprising the univariate probabilities of the city configurations for each position. One main difference of this hybrid MEDA/D with our approach, and with the proposal of Zhou et al [21], is that a single matrix is learned using all the solutions. Therefore, the information contained in the univariate model combines information from all the subproblems, disregarding the potential regularities contained in the local neighborhoods. Furthermore, since the sampling process from the matrix does not take into account the constraints related to the permutations, repair mechanisms and penalty functions are used to "correct" the infeasible solutions. As a consequence, much of the information sampled from the model to the solution can be modified by the application of the repair mechanism.

These previous MEDA/Ds were applied for solving permutation-based MOPs. The application of EDAs for solving permutation problems is increasing in interest. A number

of EDAs have been also specifically designed to deal with permutation-based problems [45], [46]. The framework proposed in this paper is only investigated for binary problems. However, EDAs that deal with the permutation space can be incorporated into the framework in the future.

In [22], the authors proposed a univariate MEDA/D for solving the (binary) multi-objective knapsack problem (MOKP) that uses an adaptive operator at the sampling step to preserve diversity, i.e., prevents the learned probability vector from premature convergence. Therefore, the sampling step depends on both the univariate probabilistic vector and an extra parameter " r ".

B. MOEA/D using multivariate EDAs

In [23], a decomposition-based algorithm is proposed to solve many-objective optimization problems. The proposed framework (MACE-gD) involves two ingredients: (i) a concept called generalized decomposition, in which the decision maker can guide the underlying search algorithm toward specific regions of interest, or the entire Pareto front and (ii) an EDA based on low-order statistics, namely the cross-entropy method [47]. MACE-gD is applied on a set of many-objective continuous functions. The obtained results showed that the proposed algorithm is competitive with the standard MOEA/D and RM-MEDA [48]. The class of low-order statistics used by MACE-gD (Normal univariate models) limit the ability of the algorithm to capture and represent interactions between the variables. The univariate densities are updated using an updating rule as the one originally proposed for the PBIL algorithm [28].

In [24], the covariance matrix adaptation evolution strategy (CMA-ES) [49] is used as the probabilistic model of MOEA/D. Although CMA-ES was introduced and has been developed in the context of evolutionary strategies, it learns a Gaussian model of the search. The covariance matrix learned by CMA-ES is able to capture dependencies between the variables. However, the nature of probabilistic modeling in the continuous domain is different to the one in the discrete domain. The methods used for learning and sampling the models are different. Furthermore, the stated main purpose of the work presented in [24] was *to investigate to what extent the CMA-ES could be appropriately integrated in MOEA/D and what are the benefits one could obtain*. Therefore, emphasis was put on the particular adaptations needed by CMA-ES to efficiently learn and sample its model in this different context. Since these adaptations are essentially different that the ones required by the discrete EDAs used in this paper, the contributions are different.

C. Other MO-EDAs

Pelikan et al. [17] discussed the multi-objective decomposable problems and their difficulty. The authors attempted to review a number of MO-EDAs, such as: multi-objective mixture-based iterated density estimation algorithm (mMIDEA) [50], multi-objective mixed Bayesian optimization algorithm (mmBOA) [51] and multi-objective hierarchical BOA (mohBOA) [52]. Moreover, the authors introduced an improvement to

mohBOA. The algorithm combines three ingredients: (i) the hierarchical Bayesian optimization algorithm (hBOA) [52], (ii) the multi-objective concepts from NSGAI [53] and (iii) clustering in the objective space. The experimental study showed that the mohBOA efficiently solved multi-objective decomposable problems with a large number of competing *building blocks*. The algorithm was capable of effective recombination by building and sampling Bayesian networks with decision trees, and significantly outperformed algorithms with standard variation operators on problems that require effective linkage learning.

All MO-EDAs covered in [17] are Pareto-based and they use concepts from algorithms such as NSGAI and SPEA2 [51]. Since, in the past few years, the MOEA/D framework has been one of the major frameworks to design MOEAs [5], incorporating probabilistic graphical models into MOEA/D seems to be a promising technique to solve scalable deceptive multi-objective problems.

Martins et al. [43] proposed a new approach for solving decomposable deceptive multi-objective problems. The MO-EDA, called *moφGA*, uses a probabilistic model based on a phylogenetic tree. The *moφGA* was tested on the multi-objective deceptive functions f_{trap5} and f_{inv_trap5} . The *moφGA* outperformed mohBOA in terms of number of function evaluations to achieve the exact *PF*, specially when the problem in question increased in size.

A question discussed in [43] is: if such probabilistic model can identify the correct correlation between the variables of a problem, the combination of improbable values of variables can be avoided. However, as the model becomes more expressive, the computational cost of incurred by the algorithm to build the model also grows. Thus, there is a trade-off between the efficiency of the algorithm for building models and its accuracy. In our proposal, at each generation, N probabilistic graphical models are kept. Therefore, a higher number of subproblems N can be a drawback for the proposal in terms of efficiency (time consuming). However, if the proposed MOEA/D-GM has an adequate commitment between the efficiency and accuracy (approximated *PF*) then it is expected to behave satisfactorily.

D. Contributions with respect to previous work

We summarize some of the main contributions of our work with respect to the related research.

- We use, for the first time, a probabilistic graphical model within MOEA/D to solve combinatorial MOPs. In this case, the previous MOEA/Ds that incorporate probabilistic models cover only univariate models.
- We investigate a particular class of problems (deceptive MOPs) for which there exist extensive evidence about the convenience of using probabilistic graphical models.
- We introduced in MOEA/D the class of probabilistic models where the structure of the model is learned from the neighborhood of each solution.
- We investigate the question of how the problem interactions are kept in the scalarized subproblems and how these interactions are translated to the probabilistic models.

VIII. EXPERIMENTS

The proposed MOEA/D-GM framework has been implemented in C++. In the comparison study, four different instantiations are used. For convenience, we called each algorithm instantiated as: MOEA/D-GA, MOEA/D-UMDA, MOEA/D-PBIL and MOEA/D-Tree.

To evaluate the algorithms, the deceptive functions f_{trap5} and f_{inv_trap5} (Section VI) are used to compose the bi-objective *Trap* problem with different number of variables $n \in \{30, 50, 100\}$. Equation (9) defines the notation of the *bi-Trap* function.

$$bi-Trap(\mathbf{x}) = (f_{trap5}(\mathbf{x}), f_{inv_trap5}(\mathbf{x})) \quad (9)$$

subject to $\mathbf{x} \in \{0, 1\}^n$

A. Performance metrics

The *true PF* from *bi-Trap* is known. Therefore, two performance metrics are used to evaluate the algorithms performance: (i) inverted generational distance metric (IGD) [54], and (ii) the number of *true* Pareto optimal solutions found by the algorithm through the generations. The stop condition for all the algorithms is $5n$ generations.

Let P^* be a set of uniformly distributed Pareto optimal solutions along the *true PF* in the objectives. P be an **approximated** set to the *true PF* obtained by an algorithm.

IGD-metric [54]: The inverted generational distance from P^* to P is defined as

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \quad (10)$$

where $d(v, P)$ is the minimum Euclidean distance between v and the solutions in P . If $|P^*|$ is large enough $IGD(P^*, P)$ could measure both convergence and diversity of P in a sense. The lower the $IGD(P^*, P)$, the better the approximation of P^* to the *true PF*.

Number of true Pareto optimal solutions: The number of Pareto solutions that composes P that belongs to the P^* is defined as

$$|P^+| = |P^* \cap P| \quad (11)$$

where $|P^+|$ is the cardinality of P^+ .

Also, we use the statistical test *Kruskall-Wallis* [55] to rank the algorithms according to the results obtained by the metrics. If two or more algorithms achieve the same rank, it means that there is no significant difference between them.

B. Parameters Settings

The following parameters settings were used in the experimental study:

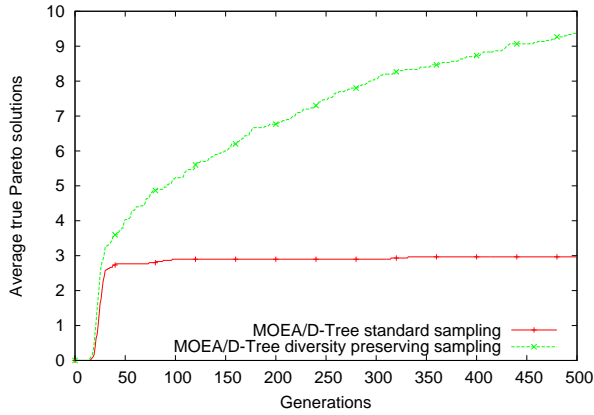
- 1) *Number of subproblems (N)*: As in most of MOEA/D algorithms proposed in the literature [2], [6], [5], the number of subproblems N and their correspondent weight vectors $\lambda^1, \dots, \lambda^N$ are controlled by a parameter H , which makes a wide spread distribution of the weight vectors according to $N = C_{H+m-1}^{m-1}$. Thus, for the

TABLE I: Results from the average IGD and average number of *true* Pareto optimal solutions $|P^+|$ computed from the 30 runs for each combination (problem size $n \times$ Algorithm \times preserving sampling). The column $|P^*|$ is the total number of *true* Pareto solutions for each problem n .

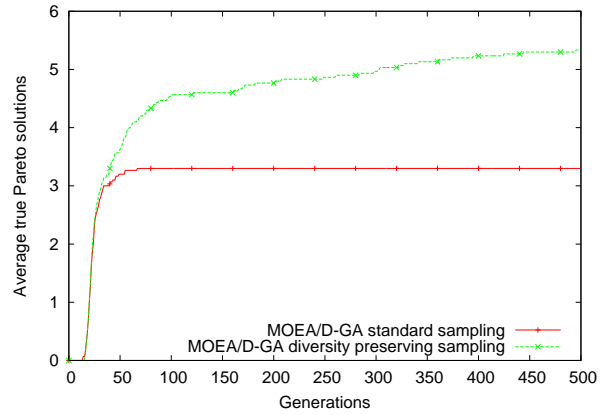
n	$ P^* $	Average number of true Pareto optimal solutions $ P^+ $							
		standard sampling				diversity preserving sampling (ds)			
		MOEA/D-GA	MOEA/D-PBIL	MOEA/D-UMDA	MOEA/D-Tree	MOEA/D-GA	MOEA/D-PBIL	MOEA/D-UMDA	MOEA/D-Tree
30	7	4.167	3.034	2.767	6.067	6.267	4.034	4.134	6.9
50	11	3.867	2.567	3.034	5.134	6.634	2.867	3.2	9.5
100	21	3.3	2.534	3.5	2.967	5.334	2.8	3.433	9.367
average IGD measure $IGD(P^*, P)$									
30	7	1.076	1.764	1.853	0.418	0.371	1.258	1.246	0.053
50	11	2.174	3.585	3.229	1.657	1.308	3.4	3.158	0.501
100	21	4.757	7.663	6.38	4.815	4.142	7.473	6.726	3.359

TABLE II: Kruskal Wallis statistical ranking test according to the IGD and $|P^+|$ for each combination (problem size $n \times$ Algorithm \times preserving sampling). The first value is the average rank over the 240 runs (i.e., 30 runs \times 8 algorithms). The second value (in brackets) is the final ranking from 8 to 1. If two or more ranks are the same, it means that there is no significant difference between them.

n	Kruskal-Wallis ranking - number of true Pareto optimal solutions $ P^+ $							
	MOEA/D-GA	MOEA/D-PBIL	MOEA/D-UMDA	MOEA/D-Tree	MOEA/D-GA	MOEA/D-PBIL	MOEA/D-UMDA	MOEA/D-Tree
30	139.37 (5.50)	184.88 (6.00)	195.72 (6.50)	66.25 (2.00)	59.43 (2.00)	146.68 (6.00)	141.87 (6.00)	29.80 (2.00)
50	126.48 (5.00)	188.30 (6.50)	165.55 (6.00)	88.15 (3.00)	52.65 (2.00)	171.40 (6.00)	153.43 (6.00)	18.03 (1.50)
100	136.43 (5.50)	178.67 (6.00)	120.37 (5.00)	151.77 (5.50)	61.58 (2.00)	162.40 (5.50)	128.37 (5.50)	24.42 (1.50)
n	Kruskal-Wallis ranking - IGD measure							
	MOEA/D-GA	MOEA/D-PBIL	MOEA/D-UMDA	MOEA/D-Tree	MOEA/D-GA	MOEA/D-PBIL	MOEA/D-UMDA	MOEA/D-Tree
30	134.13 (5.50)	187.05 (6.00)	195.22 (6.50)	65.32 (2.00)	59.43 (2.00)	148.67 (6.00)	144.38 (6.00)	29.80 (2.00)
50	106.88 (3.50)	194.43 (6.50)	169.37 (6.50)	74.47 (2.50)	58.33 (2.50)	179.35 (6.50)	160.17 (6.00)	21.00 (2.00)
100	80.30 (2.50)	204.03 (7.00)	145.72 (6.00)	82.08 (2.50)	55.45 (2.50)	195.37 (6.50)	162.12 (6.50)	38.93 (2.50)



(a) MOEA/D-Tree



(b) MOEA/D-GA

Fig. 1: Average *true* Pareto optimal solutions $|P^+|$ over the generations. Each chart shows a comparison between the standard sampling and the diversity preserving sampling using the same algorithm.

bi-objective problem, we set $H = 200$, consequently $N = 201$.

- 2) *Neighborhood size* (T_s): As the number of selected solutions is crucial for EDAs, we test a range of neighborhood size values $T_s = (1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150)$. Also, as in [3], $T_s = T_r$.
- 3) *Maximal number of replacements by a new solution* (n_r): As in [6], $n_r = 2$.
- 4) *Scalarization function*: We have applied both *Weighted Sum* and *Tchebycheff* approaches. As both achieve very similar results, only the results with *Tchebycheff* are presented in this paper.
- 5) *Maximum number of tries in diversity preserving sam-*

pling procedure: We set this parameter with the same value of the neighborhood size T_s .

- 6) *Genetic operators from MOEA/D-GA*: uniform crossover and mutation rate $= \frac{1}{n}$.
- 7) *PBIL learning rate*: $\alpha = 0.05$.

Each combination of (algorithm \times parameters setting) is independently run 30 times for each problem size n .

C. Comparison of the different variants of MOEA/D-GM for the bi-Trap5

First, the algorithms are evaluated using a fixed neighborhood size value ($T_s = 20$). Table I presents the average values of the IGD and the number of *true* Pareto solutions $|P^+|$ computed using the approximated *PF* found by the algorithms.

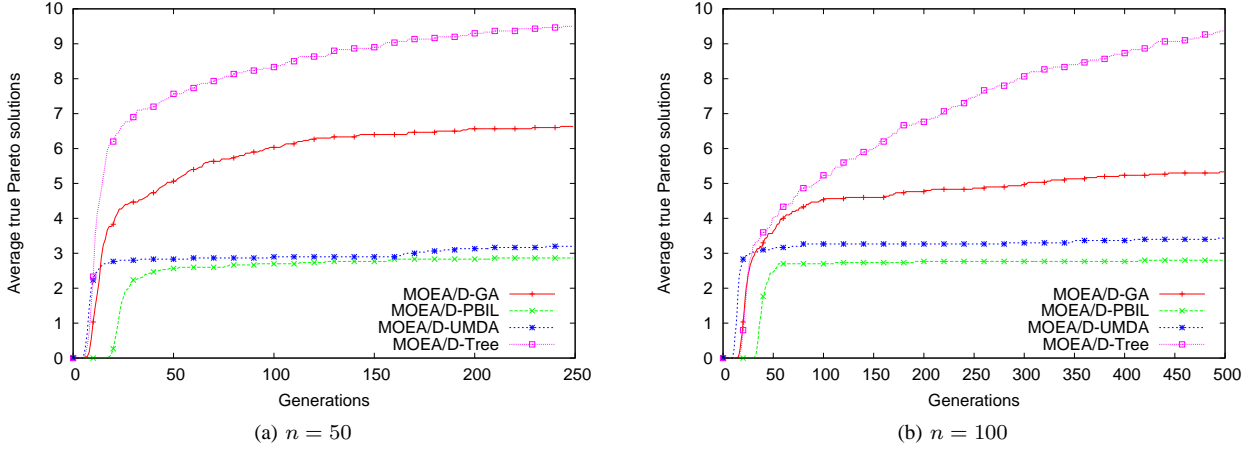


Fig. 2: Average *true* Pareto solutions $|P^+|$ through the generations for problem size 50 and 100 with $T_s = 20$

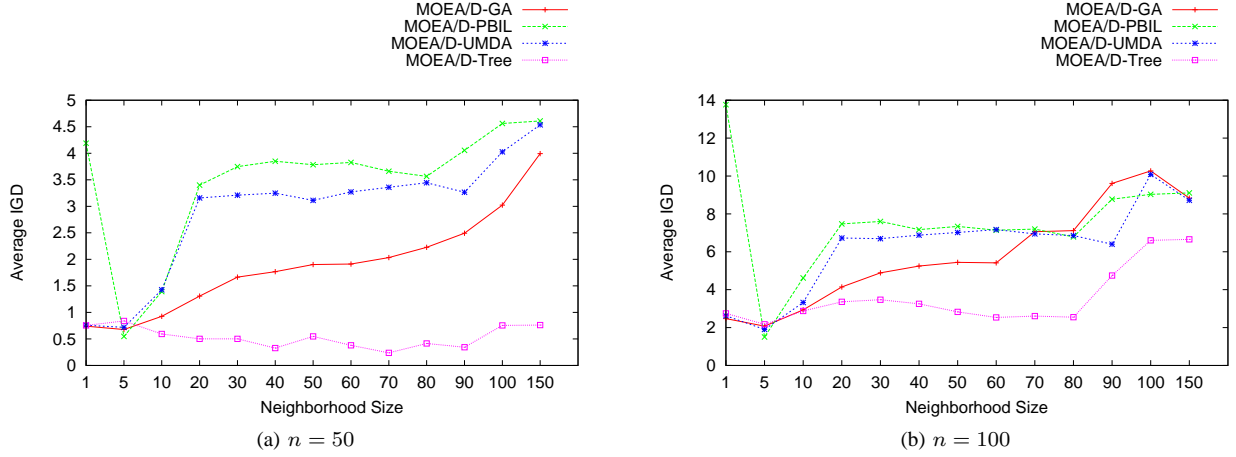


Fig. 3: Average IGD with different neighborhood sizes for selection $T_s \in \{1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150\}$

Table II presents the results of the *Kruskal-Wallis* test (at 5% significance level) applied to the results obtained by the algorithms (the same results summarized in Table I). The best rank algorithm(s) is(are) shown in bold. Figure 1 and 2 show the behavior of the algorithms throughout the generations according to the average $|P^+|$ obtained.

From the analysis of the results, we can extract the following conclusions:

According to Table II, the MOEA/D-Tree that uses the diversity preserving mechanism (MOEA/D-Tree-ds) is the only algorithm that has achieved the best rank in all the cases according to both indicators. Moreover, the diversity preserving sampling has improved the behavior of all the algorithms (mainly the MOEA/D-GA and MOEA/D-Tree) in terms of the quality of the approximation to the *true PF*. Figure 1 confirms these results. Therefore, the diversity preserving sampling has a positive effect in the algorithms for the *bi-Trap5* problem, i.e., the algorithms are able to achieve a more diverse set of solutions. Additionally, Figure 2 shows that MOEA/D-Tree-ds achieves better results than the other algorithms from the first generations.

Moreover, all the algorithms can find, at least, one global optimal solution, i.e., a true Pareto solution. One possible

explanation for this behavior is a potential advantageous effect introduced by the clustering of the solutions determined by the MOEA/D framework. This benefit would be independent of the type of models used to represent the solutions. Grouping similar solutions in a neighborhood may allow the univariate algorithm to produce some global optima, even if the functions are deceptive.

D. Influence of the neighborhood size selection T_s for the learning

The neighborhood size has a direct impact in the search ability of the MOEA/Ds, which can balance convergence and diversity for the target problem. Figure 3 presents the IGD values obtained with different neighborhood sizes (T_s).

As we mentioned before, the number of selected solutions has an impact for most of the EDAs. A multi-variate EDA needs a large set of selected solutions to be able to learn dependencies between the decision variables [30]. This fact may explain (see Figure 3) that the differences between the algorithms are lower as the neighborhood size becomes smaller (e.g., $T_s = 5$). A remarkable point in these results is that, a small T_s is better to solve *bi-Trap5* for the algorithms except for MOEA/D-Tree-ds. MOEA/D-Tree-ds achieves good results

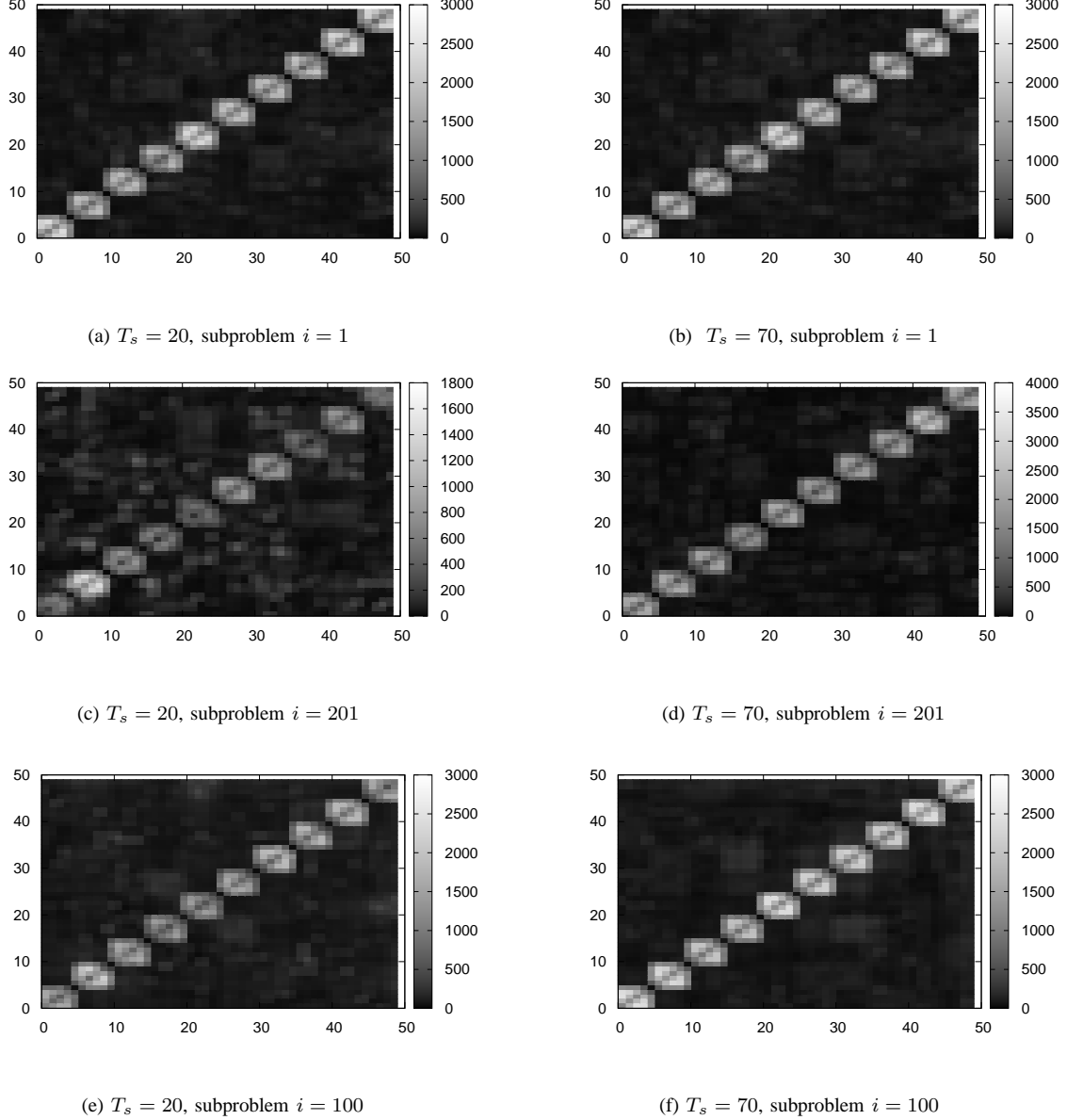


Fig. 4: Frequency matrices (*heat maps*) learned by the subproblems $i = (1, 100, 201)$ from the MOEA/D-Tree-ds for problem size $n = 50$.

with a large neighborhood size (e.g., 60, 70 and 80). Even if the neighborhood size has an important influence in the behavior of the algorithm, this parameter can not be seen in isolation of other parameters that also influence the behavior of the algorithm.

E. Analyzing the structure of problems as captured by the tree model

One of the main benefits of EDAs is their capacity to reveal a priori unknown information about the problem structure. Although this question has been extensively studied [40], [56], [57] in the single-objective domain, the analysis in the multi-objective domain are still few [19], [58], [59]. Therefore, a relevant question was to determine to what extent is the

structure of the problem captured by the probabilistic models used in MOEA/D-GM. This is a relevant question since there is no clue about the types of interactions that could be captured from models learned for scalarized functions. In this section, the structures learned from MOEA/D-Tree-ds while solving different subproblems are investigated.

In each generation, for each subproblem, a tree model is built according to the bi-variate probabilities obtained from its selected population. We can represent the tree model as a matrix $M_{n \times n}$, where each position M_{jk} represents a relationship (*pairwise*) between two variables j, k . $M_{jk} = 1$ if j is the parent of k in tree model learned, otherwise $M_{jk} = 0$.

Figure 4 represents the merge frequency matrices obtained by the 30 runs. The frequencies are represented using *heat*

maps, where lighter colors indicate a higher frequency. We have plotted the merge matrices learned from the extreme subproblems and the middle subproblem i.e., $i = (1, 100, 201)$ using the problem size $n = 50$. Because of the results of the previous section, the frequency matrices for two neighborhood sizes are presented, $T_s = (20, 70)$.

The matrices clearly show a strong relationship between the subsets of variables "building blocks" of size 5, which shows that the algorithm is able to learn the structure of the *Trap5* functions. Notice that, for neighborhood size 70, the MOEA/D-Tree-ds was able to capture more accurate structures, which exalts the good results found in accordance with the IGD metric. This can be explained by the fact that a higher population size reduces the number of spurious correlations learned from the data.

Moreover, analyzing the different scalar subproblems, we can see that, the algorithm is able to learn a structure even for the middle scalar subproblem $i = 100$, where the two conflict objectives functions f_{trap5} and f_{inv_trap5} compete in every partition (building block) of the decomposable problem.

IX. CONCLUSIONS AND FUTURE WORK

In this paper, a novel and general MOEA/D framework able to instantiate probabilistic graphical models named MOEA/D-GM has been introduced. PGMs are used to obtain a more comprehensible representation of a search space. Consequently, the algorithms that incorporate PGMs can provide a model expressing the regularities of the problem structure as well as the final solutions. The PGM investigated in this paper takes into account the interactions between the variables by learning a maximum weight spanning tree from the bi-variate probabilities distributions. An experimental study on a bi-objective version of a well known deceptive function (*bi-Trap5*) was conducted. In terms of accuracy (approximating of the true *PF*), the results have shown that the instantiation from MOEA/D-GM, called MOEA/D-Tree, is significantly better than MOEA/D that uses univariate EDAs and traditional genetic operators.

Moreover, other enhancement has been introduced in MOEA/D. A new simple but effective mechanism of sampling has been proposed, called diversity preserving sampling. Since sampling solutions already present in the neighbor solutions can have a detrimental effect in terms of convergence, the diversity preserving sampling procedure tries generating those solutions that are different from the parent solutions. According to both performance indicators, all the algorithms in the comparison have improved their results in terms of diversity to the approximation *PF*.

An analysis of the influence of the neighborhood size on the behavior of the algorithms were conducted. In general, increasing the neighborhood size has a detrimental effect. Although, this is not always the case for MOEA/D-Tree.

Also, independent of the type of models used to represent the solutions, grouping similar solutions in a neighborhood may allow to produce some global optima, even if the functions are deceptive.

Finally, we have also investigated for the first time to what extent the models learned by MOEA/D-GMs can capture

the structure of the objective functions. The analysis has been conducted for MOEA/D-Tree-ds considering the structures learned with different neighborhood sizes and different subproblems. We have found that even if relatively small neighborhoods are used (in comparison with the standard population sizes used in EDAs), the models are able to capture the interactions of the functions. One potential application of this finding, is that, we could reuse or transfer models between subproblems, in a similar way to the application of structural transfer between related problems [60].

The scalability of MOEA/D-Tree-ds for the *bi-Trap5* problem and other multi-objective deceptive functions should be investigated. The "most appropriate" model for *bi-Trap5* should be able to learn higher order interactions (of order 4). Therefore, PGMs based on Bayesian or Markov networks could be of application in this case.

The directions for future work are: (i) Conceive strategies to avoid learning a model for each subproblem. This would improve the results in terms of computational cost; (ii) Use of the most probable configurations of the model to speed up convergence; (iii) Consider the application of hybrid schemes incorporating local search that take advantage of the information learned by the models; (iv) Evaluate MOEA/D-GM on other benchmarks of deceptive MOPs.

ACKNOWLEDGMENTS

This work has received support by CNPq (Productivity Grant Nos. 305986/2012-0 and Program Science Without Borders Nos.: 400125/2014-5), by CAPES (Brazil Government), by IT-609-13 program (Basque Government) and TIN2013-41272P (Spanish Ministry of Science and Innovation).

REFERENCES

- [1] C. C. Coello, G. Lamont, and D. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems.*, 2nd ed., ser. Genetic and Evolutionary Computation. Berlin, Heidelberg: Springer, 2007.
- [2] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [3] Z. Wang, Q. Zhang, G. Maoguo, and Z. Aimin, "A Replacement Strategy for Balancing Convergence and Diversity in MOEA/D," in *Proceedings of the Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 2132–2139.
- [4] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE Transaction Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [5] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evolutionary Computation*, no. 5, pp. 694–716, 2015.
- [6] H. Li and Q. Zhang, "Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II," *IEEE Transaction on Evolutionary Computation*, pp. 284–302, 2008.
- [7] H. Ishibuchi, N. Akedo, and Y. Nojima, "A Study on the Specification of a Scalarizing Function in MOEA/D for Many-Objective Knapsack Problems," in *Proceedings of the 7th International Learning and Intelligent Optimization Conference (LION)*, ser. Lecture Notes in Computer Science, vol. 7997. Springer, 2013, pp. 231–246.
- [8] Y.-y. Tan, Y.-c. Jiao, H. Li, and X.-k. Wang, "MOEA/D+ uniform design: A new version of MOEA/D for optimization problems with many objectives," *Computers & Operations Research*, vol. 40, no. 6, pp. 1648–1660, 2013.
- [9] A. J. Nebro and J. J. Durillo, "A study of the parallelization of the multi-objective metaheuristic MOEA/D," in *Learning and Intelligent Optimization*. Springer, 2010, pp. 303–317.

- [10] M. Pilat and R. Neruda, "Incorporating user preferences in MOEA/D through the coevolution of weights," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. ACM, 2015, pp. 727–734.
- [11] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary computation*, vol. 22, no. 2, pp. 231–264, 2014.
- [12] A. Alhindi and Q. Zhang, "MOEA/D with tabu search for multiobjective permutation flow shop scheduling problems," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014*. IEEE, 2014, pp. 1155–1164.
- [13] L. Ke, Q. Zhang, and R. Battiti, "Hybridization of decomposition and local search for multiobjective optimization," *IEEE T. Cybernetics*, vol. 44, no. 10, pp. 1808–1820, 2014.
- [14] P. Larrañaga, H. Karshenas, C. Bielza, and R. Santana, "A review on probabilistic graphical models in evolutionary computation," *Journal of Heuristics*, vol. 18, no. 5, pp. 795–819, 2012.
- [15] J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, Eds., *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*. Springer, 2006.
- [16] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I. Binary parameters," in *Parallel Problem Solving from Nature - PPSN IV*, ser. Lectures Notes in Computer Science, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds., vol. 1141. Berlin: Springer, 1996, pp. 178–187.
- [17] M. Pelikan, K. Sastry, and D. E. Goldberg, "Multiobjective estimation of distribution algorithms," in *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, ser. Studies in Computational Intelligence, M. Pelikan, K. Sastry, and E. Cantú-Paz, Eds. Springer, 2006, pp. 223–248.
- [18] L. Martí, J. García, A. Berlanga, and J. M. Molina, "Multi-objective optimization with an adaptive resonance theory-based estimation of distribution algorithm: A comparative study," in *LION*, ser. Lecture Notes in Computer Science, C. A. C. Coello, Ed., vol. 6683. Springer, 2011, pp. 458–472.
- [19] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga, "Multi-objective optimization based on joint probabilistic modeling of objectives and variables," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 519–542, 2014.
- [20] V. A. Shim, K. C. Tan, and K. K. Tan, "A Hybrid Estimation of Distribution Algorithm for Solving the Multi-objective Multiple Traveling Salesman Problem," in *Proceedings of the Congress on Evolutionary Computation (CEC)*. IEEE, 2012, pp. 1–8.
- [21] A. Zhou, F. Gao, and G. Zhang, "A decomposition based estimation of distribution algorithm for multiobjective traveling salesman problems," *Computers and Mathematics with Applications*, vol. 66, no. 10, pp. 1857–1868, 2013.
- [22] B. Wang, X. Hua, and Y. Yuan, "Scale Adaptive Reproduction Operator for Decomposition based Estimation of Distribution Algorithm," in *Proceedings of the Congress on Evolutionary Computation (CEC)*. IEEE, 2015, pp. 1–8.
- [23] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition and cross entropy methods for many-objective optimization," *Inf. Sci.*, vol. 282, pp. 363–387, 2014.
- [24] S. Zapotecas-Martínez, B. Derbel, A. Liefooghe, D. Brockhoff, H. Aguirre, and K. Tanaka, "Injecting CMA-ES into MOEA/D," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. ACM, 2015, pp. 783–790.
- [25] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition," in *7th International Conference Evolutionary on Multi-Criterion Optimization EMO*, ser. Lecture Notes in Computer Science, vol. 7811. Springer, 2013, pp. 428–442.
- [26] M. Pelikan, "Analysis of epistasis correlation on NK landscapes with nearest neighbor interactions," in *Proceedings of the 13th conference on Genetic and evolutionary computation (GECCO)*. ACM, 2011, pp. 1013–1020.
- [27] H. Mühlenbein, "The equation for response to selection and its use for prediction," *Evolutionary Computation*, vol. 5, no. 3, pp. 303–346, 1998.
- [28] S. Baluja, "Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-CS-94-163, 1994.
- [29] M. Zangari, R. Santana, A. Pozo, and A. Mendiburu, "Not all PBILs are the same: Unveiling the different learning mechanisms of PBIL variants," 2015, submitted to publication.
- [30] H. Mühlenbein, T. Mahnig, and A. Ochoa, "Schemata, distributions and graphical models in evolutionary optimization," *Journal of Heuristics*, vol. 5, no. 2, pp. 213–247, 1999.
- [31] M. Pelikan and H. Mühlenbein, "The bivariate marginal distribution algorithm," in *Advances in Soft Computing - Engineering Design and Manufacturing*, R. Roy, T. Furuhashi, and P. Chawdhry, Eds. London: Springer, 1999, pp. 521–535.
- [32] R. Santana, A. Ochoa, and M. R. Soto, "The mixture of trees factorized distribution algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO*. San Francisco, CA: Morgan Kaufmann Publishers, 2001, pp. 543–550.
- [33] S. Baluja and S. Davies, "Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space," in *Proceedings of the 14th International Conference on Machine Learning*, D. H. Fisher, Ed., 1997, pp. 30–38.
- [34] T. Mahnig and H. Mühlenbein, "Optimal mutation rate using bayesian priors for estimation of distribution algorithms," in *SAGA*, ser. Lecture Notes in Computer Science, K. Steinhofel, Ed., vol. 2264. Springer, 2001, pp. 33–48.
- [35] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "BOA: The Bayesian optimization algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, vol. I. Orlando, FL: Morgan Kaufmann Publishers, San Francisco, CA, 1999, pp. 525–532.
- [36] S. Shakyia and R. Santana, Eds., *Markov Networks in Evolutionary Computation*. Springer, 2012.
- [37] D. E. Goldberg, "Simple genetic algorithms and the minimal, deceptive problem," in *Genetic Algorithms and Simulated Annealing*, L. Davis, Ed. London, UK: Pitman Publishing, 1987, pp. 74–88.
- [38] K. Deb and D. E. Goldberg, "Analyzing deception in trap functions," University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, IlliGAL Report 91009, 1991.
- [39] K. Sastry, H. A. Abbass, D. E. Goldberg, and D. Johnson, "Substructural niching in estimation of distribution algorithms," in *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*. ACM, 2005, pp. 671–678.
- [40] C. Echegoyen, J. A. Lozano, R. Santana, and P. Larrañaga, "Exact Bayesian network learning in estimation of distribution algorithms," in *Proceedings of the Congress on Evolutionary Computation CEC*. IEEE Press, 2007, pp. 1051–1058.
- [41] C. Echegoyen, A. Mendiburu, R. Santana, and J. A. Lozano, "Toward understanding EDAs based on Bayesian networks through a quantitative analysis," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 173–189, 2012.
- [42] M. Pelikan, K. Sastry, and D. E. Goldberg, "Multiobjective hBOA, clustering and scalability," University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, IlliGAL Report No. 2005005, February 2005.
- [43] J. P. Martins, A. H. M. Soares, D. V. Vargas, and A. C. B. Delbem, "Multi-objective phylogenetic algorithm: Solving multi-objective decomposable deceptive problems," in *Evolutionary Multi-Criterion Optimization*. Springer, 2011, pp. 285–297.
- [44] M. Pelikan, *Hierarchical Bayesian Optimization Algorithm. Toward a New Generation of Evolutionary Algorithms*, ser. Studies in Fuzziness and Soft Computing. Springer, 2005, vol. 170.
- [45] J. Ceberio, E. Irurozki, A. Mendiburu, and J. A. Lozano, "A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 103–117, 2012.
- [46] J. Ceberio, A. Mendiburu, and J. A. Lozano, "The Plackett-Luce ranking model on permutation-based optimization problems," in *Proceedings of the Congress on Evolutionary Computation (CEC)*, IEEE. IEEE press, 2013, pp. 494–501.
- [47] Z. Botev, D. P. Kroese, R. Y. Rubinstein, P. L'Ecuyer et al., *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-carlo Simulation*. North-Holland Mathematical Library-Elsevier, 2013.
- [48] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.
- [49] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1996, pp. 312–317.
- [50] D. Thierens and P. A. Bosman, "Multi-objective optimization with iterated density estimation evolutionary algorithms using mixture models," in *Evolutionary Computation and Probabilistic Graphical Models*.

- Proceedings of the Third Symposium on Adaptive Systems (ISAS-2001)*, Cuba, Havana, Cuba, March 2001, pp. 129–136.
- [51] M. Laumanns and J. Ocenasek, “Bayesian optimization algorithm for multi-objective optimization,” in *Proceedings of the 7th Parallel Problem Solving from Nature Conference (PPSN)*, ser. Lecture Notes in Computer Science, vol. 2439. Springer, 2002, pp. 298–307.
 - [52] N. Khan, “Bayesian optimization algorithms for multi-objective and hierarchically difficult problems,” Master’s thesis, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, 2003.
 - [53] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable multi-objective optimization test problems,” in *Proceedings of the Congress on Evolutionary Computation (CEC)*, vol. 2. IEEE press, 2002, pp. 825–830.
 - [54] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
 - [55] J. Derrac and S. Garcia, “A Practical Tutorial on the use of Nonparametric Statistical Tests as a Methodology for Comparing Evolutionary and Swarm Intelligence Algorithms,” *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
 - [56] A. E. I. Brownlee, J. McCall, and M. Pelikan, “Influence of selection on structure learning in Markov network EDAs: An empirical study,” in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO*. ACM, 2012, pp. 249–256.
 - [57] R. Santana, P. Larrañaga, and J. A. Lozano, “Protein folding in simplified models with estimation of distribution algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 4, pp. 418–438, 2008.
 - [58] G. Fritsche, A. Strickler, A. Pozo, and R. Santana, “Capturing relationships in multi-objective optimization,” in *Proceedings of the Brazilian Conference on Intelligent Systems (BRACIS 2015)*, Natal Brazil, 2015, accepted for publication.
 - [59] R. Santana, C. Bielza, J. A. Lozano, and P. Larrañaga, “Mining probabilistic models learned by EDAs in the optimization of multi-objective problems,” in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO*. ACM, 2009, pp. 445–452.
 - [60] R. Santana, A. Mendiburu, and J. A. Lozano, “Structural transfer using EDAs: An application to multi-marker tagging SNP selection,” in *Proceedings of the 2012 Congress on Evolutionary Computation (CEC)*. IEEE Press, 2012, pp. 3484–3491, best Paper Award of 2012 Congress on Evolutionary Computation.